

选题 5.15 技术报告

未央-软件 01 贾清炜 2020013061

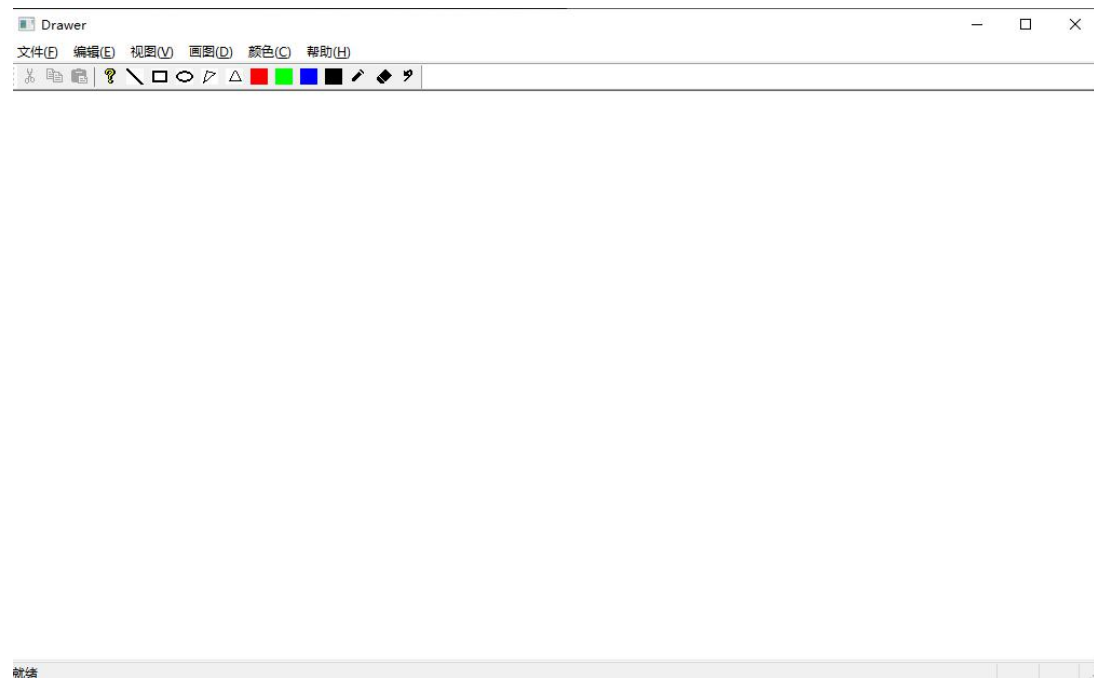
一、作业要求

题目 5.15 【**】画图工具

本次课程设计的任务是利用 VC 开发环境和 MFC 编写一个简单的画图工具。要求该工具可以支持 bmp 或 jpg 或 png 等至少 1 中常用的图像文件格式，即可以创建和打开这种格式的图像文件，并在此基础上可以生成和编辑线段、矩形、三角形和圆等基本图形，同时可以将编辑结果以这种格式保存到文件当中。

二、图形界面及功能介绍

①图形界面



②功能详解

(1) 文件 (F)

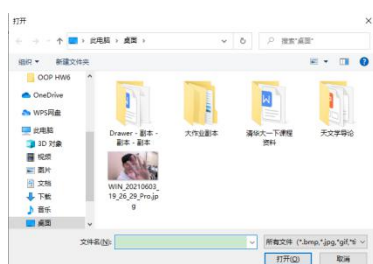
(I) 菜单展开图例



（II）功能介绍

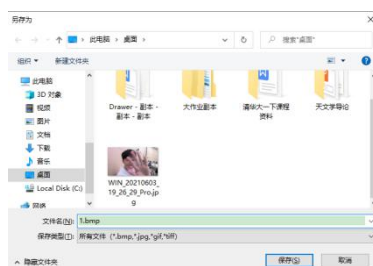
1. 打开（O） Ctrl+O

点击“打开”按钮后可以打开如下界面，选择 JPG 或 BMP 格式的文件，并点击打开后能够将文件投到程序界面中，进行编辑操作。



2. 另存为（S） Ctrl+S

点击“另存为”按钮后可以打开如下界面，将当前程序界面编辑得到的图像输出为 JPG 或 BMP 格式的文件（使用时需要输入完整的文件名，并加上后缀，如“1.jpg”“1.bmp”等）。



3. 退出（X） Ctrl+X

点击“退出”按钮后可以退出程序。

（2）编辑（E）

（I）菜单展开图例



（II）功能介绍

1. 撤销（Z） Ctrl+Z

点击“撤销”按钮后可以撤销最后一个由 Drawer 程序编辑的图案。

说明：快捷栏中最后一个快捷按钮，即样式的按钮可以实现相同的功能。

（3）视图（V）

（I）菜单展开图例

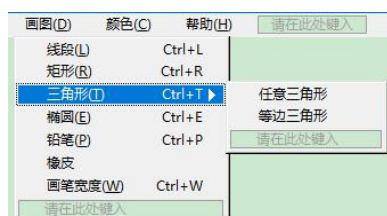


（II）功能介绍

视图类下属按钮均为 MFC 创建过程中自带系统，此文档不进行具体说明。

（4）画图（D）

（I）菜单展开图例



（II）功能介绍

1. 线段（L） Ctrl+L

点击“线段”按钮可以将绘制方式调整为线段模式，实现方式为在鼠标左键按下时所处点与左键抬起时所在点之间连接一条直线，实现对线段的绘制。

说明：快捷栏中第二个按钮，即样式的按钮可以实现同样的功能。

2. 矩形（R） Ctrl+R


点击“矩形”按钮可以将绘制方式调整为矩形模式，实现方式为将鼠标左键按下时所处点与左键抬起时所在点分别作为矩形两个相对的顶角，实现对矩形的绘制。通过在绘制过程中按


下 **Ctrl** 按钮可以实现将所绘制矩形转变为边长为短边的正方形。

说明：快捷栏中第三个按钮，即  样式的按钮可以实现同样的功能。

3. 三角形 (T) **Ctrl+T**

点击“三角形”按钮可以选择“任意三角形”和“等边三角形”两种绘制模式。其中，任意三角形绘制模式可以绘制任意一个三角形，实现方式为在操作界面上点三个不同的点，程序连接三点绘制成一个三角形。等边三角形绘制模式可以以鼠标左键按下作为等边三角形的额中心、以抬起时所在点（或其对称点）作为等边三角形的一个底角绘制向上的等边三角形。

说明：快捷栏中第五个按钮，即  样式的按钮可以实现与任意三角形绘制模式同样的功能；

快捷栏中第六个按钮，即  样式的按钮可以实现与等边三角形绘制模式同样的功能。

4. 椭圆 (E) **Ctrl+E**

点击“椭圆”按钮可以选择椭圆绘制模式，实现方式为在鼠标左键按下所在点和抬起时所在点为相对顶点的方框内绘制椭圆形。通过在绘制过程中按下 **Ctrl** 按钮可以实现将所绘制的椭圆形变为半径为椭圆半短轴长的圆形。

说明：快捷栏中第四个按钮，即  样式的按钮可以实现相同的功能。

5. 铅笔 (P) **Ctrl+P**

点击“铅笔”按钮可以选择铅笔绘制模式，实现方式为按下鼠标左键并拖动，在程序界面中绘制出鼠标移动的轨迹。

说明：快捷栏中第十一个按钮，即  样式的按钮可以实现相同的功能。

6. 橡皮

点击“橡皮”按钮可以选择橡皮模式，实现方式为按下鼠标左键并拖动，在程序界面中擦除鼠标经过处的线条。

说明：快捷栏中第十二个按钮，即  样式的按钮可以实现相同的功能。

存在问题：由于重绘空心椭圆和长方形过程中调用了反色笔，因此会覆盖掉原先在这两类图形上的橡皮痕迹。但目前本人没有找到很好的方法解决这一问题。

7. 画笔宽度 (W) Ctrl+W



点击画笔宽度按钮会弹出式样的对话框，在对话框中输入宽度值，可以将画笔的宽度调整到需要的值。

(5) 颜色 (C)


(I) 菜单展开图例



(II) 功能介绍


1. 蓝色

点击“蓝色”按钮可以将当前画笔的颜色调整为蓝色。

说明：点击快捷栏上第九个按钮，即样式的按钮可以实现同样的功能。

2. 绿色

点击“绿色”按钮可以将当前画笔的颜色调整为绿色。

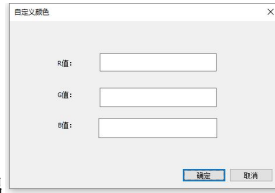
说明：点击快捷栏上第八个按钮，即样式的按钮可以实现同样的功能。

3. 红色

点击“红色”按钮可以将当前画笔的颜色调整为红色。

说明：点击快捷栏上第七个按钮，即样式的按钮可以实现同样的功能。

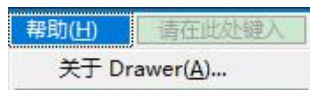
4. RGB 自选



点击“RGB 自选”按钮会弹出样式的对话框，通过分别填入需要颜色的 RGB 数值，可以将当前画笔的颜色调整对 RGB 值所对应的颜色。

（6）帮助（H）

（I）菜单展开图例

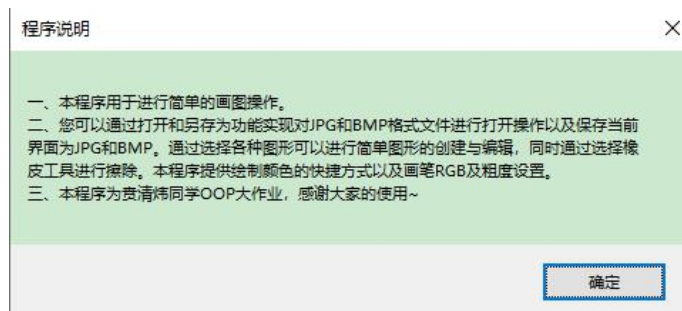


（II）功能介绍

1. 关于 Drawer（A）

点击“关于 Drawer”按钮会弹出程序的介绍窗口（窗口图样如下）。

说明：快捷栏中的第一个按钮，即  样式的按钮可以实现同样的功能。



窗口图样：

三、各画图按钮原理及验证（图形的绘制、文件的打开与保存均在 CChildView 中实现，部分定义的类在文档中第一次被应用时做出说明）

在文件 CChildView 中定义类 CChildView，定义代码如下（各变量、函数用途通过注释方式已经给出）：

```
class CChildView : public CWnd
{
// 构造
public:
```

```

CChildView();

// 特性
public:

    CString BmpName;//打开的 bmp/jpg 文件的文件名
    CString EntName;//打开的 bmp/jpg 文件的拓展名
    CBitmap m_bitmap;//用于将打开的图像文件绘制到程序中
    CPoint p;//用于储存鼠标按下时所在的坐标点位置
    CPoint term;//用于储存鼠标上次移动时所在的标点位置
    CPoint mid;//用于记录任意三角形第二个顶点的位置

    CPtrArray ptr;//用于记录已绘图形的信息
    CPen cho_pen[100];//记录每一个使用过的笔，方便重绘时调用
    COLORREF pen_color;

    CPen w_cho_pen; //用于橡皮绘制的笔

    CDC bufferdc;

    CBitmap bufferbmp;//用于绘制打开的图像文件

    int flag;//用于记录当前的绘图模式
    int do_flag;//用于判断当前
    int trnum;//记录当前任意三角形已经确定顶点的个数
    int a;
    int pen_size; //当前画笔的宽度
    int number;//记录已绘制的图像个数

// 操作
public:

    void showBitmap(CDC* pDC, CString BmpName);//展示 bmp 类型的文件
    BOOL showJpgmap(CDC* pDC, CString strPath);//展示 jpg 类型的文件
    BOOL CreatePen(int pensize, COLORREF pencolor);//创造新的画笔

```

```

// 重写

protected:

    virtual BOOL PreCreateWindow(CREATESTRUCT& cs); //MFC 绘制窗口函数

public:

    virtual void OnDraw(CDC* pDC);

    virtual BOOL CChildView::OnEraseBkgnd(CDC* pDC); //重构 OnEraseBkgnd 函数

// 实现

public:

    virtual ~CChildView();

    // 生成的消息映射函数

protected:

    afx_msg void OnPaint(); //重构 OnPaint 函数

    DECLARE_MESSAGE_MAP()

public:

    afx_msg void OnMouseMove(UINT nFlags, CPoint point);

    afx_msg void OnLButtonDown(UINT nFlags, CPoint point);

    afx_msg void OnLButtonUp(UINT nFlags, CPoint point);

    afx_msg void OnLine();

    afx_msg void OnRectangle();

    afx_msg void OnOval();

    afx_msg void OnRed();

    afx_msg void OnBlack();

    afx_msg void OnBlue();

    afx_msg void OnGreen();

    afx_msg void OnChooseRgb();

    afx_msg void OnPencil();

    afx_msg void OnEraser();

    afx_msg void OnCancel();

    afx_msg void OnTtriangle();

```



```

afx_msg void OnDtriangle();

afx_msg void OnChwidth();

afx_msg void OnFileOpen();

afx_msg void OnSave();

afx_msg void OnAppAbout(); //此处分别为点击各按钮的命令响应函数
};

```

①线段

在函数 OnLButtonDown 中记录鼠标左键按下的坐标，赋值给 p，在鼠标移动过程中，不断地通过 term 记录下每次鼠标移动的坐标，在下一次移动时先通过 dc->SetROP2(R2_NOTXORPEN); 语句调用反色笔将上一次预览中绘制的线段覆盖，并通过 MoveTo 和 LineTo 函数绘制新的预览线段，预览线段的起始点为 p，终止点为当前鼠标所在位置。当鼠标左键抬起时，在 OnLButtonUp 函数中通过 MoveTo 和 LineTo 函数绘制编辑结果，并通过 new 函数创建新的 Graph 类的实例对象，将线段的绘制粗细、颜色、起末点位置记录在 Graph 类中，并通过函数 add 添加到储存图形数据的 CPtrArray 型变量 ptr 中。

附：【Graph 类】

```

class CGraph
{
public:
    CGraph(void);

    CGraph(int nDrawType, CPoint& ptOrigin, CPoint& mid, CPoint& ptEnd, COLORREF cColor, int
cSize); //用于对实例对象参数赋值

    ~CGraph(void);

public:
    int m_nDrawType; //记录图形绘制的种类（如线段、三角形等）

    CPoint m_ptOrigin; //记录图形的起始点

    CPoint m_ptEnd; //记录图形的终止点

    CPoint m_mid; //记录任意三角形绘制模式下第二个顶点

    COLORREF m_cColor; //记录绘制图形的颜色

```

```
int m_cSize;//记录绘制图形的画笔粗细  
};
```

②椭圆

在函数 `OnLButtonDown` 中记录鼠标左键按下的坐标，赋值给 `p`，在鼠标移动过程中，不断地通过 `term` 记录下每次鼠标移动的坐标，在下一次移动时先通过 `dc->SetROP2(R2_NOTXORPEN)`；语句调用反色笔将上一次预览中绘制的椭圆覆盖，并通过 `dc->Ellipse()` 语句绘制新的预览椭圆。当鼠标左键抬起时，在 `OnLButtonUp` 函数中通过 `Ellipse` 函数绘制最终的编辑图形，通过 `new` 函数创建新的 `Graph` 类的实例对象，将椭圆的绘制粗细、颜色、`Ellipse` 函数中的参数记录在 `Graph` 类中，通过 `add` 添加到 `ptr` 中。本功能中所绘制的椭圆均为绘制在以起始点和终止点为相对顶角的矩形中的椭圆。若在绘制的任意过程中按下 `Ctrl` 按钮（本过程通过判断语句 `if (nFlags & MK_CONTROL)` 处理，其中 `nFlags` 类型为 `UINT`），则会在终止点相同的方向绘制一个边长为矩形短边的正方形中的圆形。

③矩形

在函数 `OnLButtonDown` 中记录鼠标左键按下的坐标，赋值给 `p`，在鼠标移动过程中，不断地通过 `term` 记录下每次鼠标移动的坐标，在下一次移动时先通过 `dc->SetROP2(R2_NOTXORPEN)`；语句调用反色笔将上一次预览中绘制的矩形覆盖，并通过 `dc->Rectangle()` 语句绘制新的预览矩形。当鼠标左键抬起时，在 `OnLButtonUp` 函数中通过 `Rectangle` 函数绘制最终的编辑图形，通过 `new` 函数创建新的 `Graph` 类的实例对象，将矩形的绘制粗细、颜色、`Rectangle` 函数中的参数记录在 `Graph` 类中，通过 `add` 添加到 `ptr` 中。本功能中所绘制的矩形均为以起始点和终止点为相对顶角的矩形。若在绘制的任意过程中按下 `Ctrl` 按钮（本过程通过判断语句 `if (nFlags & MK_CONTROL)` 处理，其中 `nFlags` 类型为 `UINT`），则会在终止点相同的方向绘制一个边长为矩形短边的正方形。

④铅笔

在函数 `OnLButtonDown` 响应鼠标左键按下的命令后，铅笔功能开始运作，在 `OnMouseMove` 函数中每当鼠标的坐标点有更新都会在上一个和当前坐标点之间绘制一个线段。并且通过 `Graph` 类和 `add` 函数将所有的线段全部储存到 `ptr` 中。需要说明的是，由于电脑刷新速度较

快，故所得结果与直接用铅笔划线效果相似，但由于客观存在电脑刷新时间的间隔，故本软件所应用的方法不能够很好的解决抗锯齿的问题。

⑤任意三角形

在选择了任意三角形模式后，通过函数 `OnLButtonDown` 记录下鼠标连续三次落下点的坐标，在三点之间两两作为线段的端点绘制线段，组成一个三角形。通过 `new` 函数创建新的 `Graph` 类的实例对象，将该三角形的绘制粗细、颜色、三个顶点的坐标记录在 `Graph` 类中，通过 `add` 添加到 `ptr` 中。

⑥等边三角形

在函数 `OnLButtonDown` 中记录下鼠标左键按下的坐标，赋值给 `p`，在鼠标的移动过程中不断地通过 `term` 记录下每次鼠标移动的坐标，在下一次移动时通过调用反色笔将上一次绘制的等边三角形覆盖，通过数学计算算出三角形三个顶点的坐标位置，两两连线。当鼠标左键抬起时，在 `OnLButtonUp` 函数中通过绘制最终的等边三角形，通过 `new` 函数创建新的 `Graph` 类记录参数，通过 `add` 函数添加到 `ptr` 中。由于图形刷新的问题，绘制等边三角形过程中出现了缓存里在中心点处画出小三角形的情况，具体操作中通过确定图形后立刻 `Invalidate()`；解决。

⑦橡皮

橡皮的原理与铅笔功能一致，此处不做赘述。区别为将颜色更改为画板底色白色。

⑧红绿蓝三色及 RGB 自选、画笔粗细的调节。

RGB 选择的窗口应用了 MFC 中资源视图 `Dialog` 自带的功能。在 `CChildView.cpp` 文件开头定义变量 `int R,G,B`；当选择了红蓝绿黑四色的快捷按钮时，在对应的命令响应函数中将 RGB 更改为需要的数值。通过 `CPen` 类功能创建新的画笔，并且用类 `CChildView` 中的变量 `pen_color` 实时记录 RGB 值的更改结果。画笔粗细调节原理与 RGB 自选相同。值得注意的是，`dialog` 窗口只能接受到 `CString` 类型的变量，因此需要在窗口对应处理函数中将 `CString` 变量转化为 `int` 类型。

⑨撤销及图形的重绘

本程序中所有绘制的图形均保存在 ptr 中，故撤销操作为将 ptr 中最后一个 Graph 类型的变量删除。由于 MFC 本身的特性，如果不进行图形的重绘，会导致窗口大小变化以及最小化后原先图形全部消失。本程序通过 OnPaint 函数和 OnDraw 函数进行图形的重绘，首先判断 Graph 中记录的图形种类，然后根据各种情况绘制出保存过的所有图形。

⑩文件的打开与保存

本部分技术参考了

CSDN:https://blog.csdn.net/pengju007/article/details/85396890?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522162229000016780264039304%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=162229000016780264039304&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~rank_v29-1-85396890.pc_search_result_before_js&utm_term=MFC+对BMP编辑&spm=1018.2226.3001.4187

和

CSDN: <https://blog.csdn.net/friendan/article/details/38358507>

参考过程中部分抄录，并结合实际情况更改了部分。

现将参考部分代码全文摘录于“参考代码全文摘录.txt”中。

四、功能验证

说明：本部分内容较难通过文字直接表述，故采用录屏完成，请查看文件夹中“功能验证”视频文件。

五、程序的不足

受限于本人程序设计水平的局限性，本次程序仅实现了 BMP 和 JPG 格式的文件的打开与保存，对于 PNG 格式的图像文件本程序无法实现打开与保存。同时，本程序所采用的画图方式大多为使用现有的函数进行解决，存在绘图图像抗锯齿性能差的问题。相比于 Window 自带的画图软件，本软件不能做到对具体图像范围的放大缩小，没有实现油漆桶、截图等功能。这些部分均不属于本次大作业要求范围，但希望在后续的学习中我能够学习更多的面向对象程序设计的技术，对本程序进行改进。

六、大作业总结

通过本次大作业的练习，我更好地掌握了 MFC 的知识，成功制作了自己的画图软件，也更加深入的了解了面向对象的技术。本次作业参考了很多 CSDN、MSDN 的文档，再参考并学习的同时掌握了很多新的知识，积累了编写项目的经验。希望日后能够学习更多面向对象的技术，制作出更好地画图软件，并最终制作出很多优秀的软件。